

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Presented) A computer program product, tangibly embodied in an information carrier, for developing an application, the computer program product being operable to cause data processing apparatus to:
 - receive a first model in a first language, the first model defining development objects representing building blocks for developing the application, relationships among the development objects, and constraints for developing the application;
 - generate a set of intermediate objects using the first model; and
 - generate an API using the set of intermediate objects as inputs such that the API enforces the relationships and the constraints defined in the first model and enables accessing the development objects.
2. (Previously Presented) The computer program product of claim 1, further comprising instructions to convert the first model to a second model in a second language, wherein the set of intermediate objects is generated using the second model.
3. (Original) The computer program product of claim 2, wherein the second language comprises XML.

4. (Original) The computer program product of claim 1, wherein the first language comprises UML.
5. (Original) The computer program product of claim 1, wherein the set of intermediate objects comprises Java objects.
6. (Original) The computer program product of claim 1, wherein the first language comprises a customizable extension.
7. (Previously Presented) The computer program product of claim 6, wherein the customizable extension is used to implement an additional feature of the API.
8. (Previously Presented) The computer program product of claim 7, wherein the additional feature comprises an indication of a file border.
9. (Original) The computer program product of claim 1, wherein the API comprises a copy and paste operation.
10. (Currently Amended) A computer program product, tangibly embodied in an information carrier, for developing an application, the computer program product being operable to cause data processing apparatus to:

receive a first model in a first language, the first model defining development objects representing building blocks for developing the application, relationships among the development objects, and constraints for developing the application;

generate a set of intermediate objects using the first model; and

generate an XML schema using the set of intermediate objects as inputs such that the API XML schema enforces the relationships and the constraints defined in the first model and enables implementing the development objects.

11. (Previously Presented) The computer program product of claim 10, further comprising instructions to convert the first model to a second model in a second language, wherein the set of intermediate objects is generated using the second model.

12. (Original) The computer program product of claim 11, wherein the second language comprises XML.

13. (Original) The computer program product of claim 10, wherein the first language comprises UML.

14. (Original) The computer program product of claim 10, wherein the set of intermediate objects comprises Java objects.

15. (Previously Presented) The computer program product of claim 10, wherein the XML schema includes a tree based on aggregation relationships in the first model.

16. (Previously Presented) The computer program product of claim 10, wherein the XML schema includes a reference based on an association relationship in the first model.

17. (Previously Presented) The computer program product of claim 10, wherein the XML schema includes a complex type extension based on an inheritance relationship in the first model.

18. (Currently Amended) A computer program product, tangibly embodied in an information carrier, for developing an application, the computer program product being operable to cause data processing apparatus to:

receive a [[first]] data model defining development objects representing building blocks for developing the application, relationships among the development objects, and constraints for developing the application;

derive an API based on the [[first]] data model such that the API enforces the relationships and the constraints defined in the [[first]] data model; and

use the API to perform operations on the development objects.

19. (Original) The computer program product of claim 18, wherein the API comprises an interface layer, a proxy layer, and a state layer.

20. (Previously Presented) The computer program product of claim 18, wherein the operations comprise:

creating a new development object as a transient object without an existing corresponding file; and

modifying the transient object until the transient object is committed to a persistent file.

21. (Previously Presented) The computer program product of claim 20, further comprising instructions to destroy the transient object if a delete command is requested before the transient object is committed to a persistent file.

22. (Previously Presented) The computer program product of claim 20, further comprising instructions to mark the persistent file as deleted if a delete command is requested after the transient object is committed to a persistent file.